# WinCVS 日常使用指南

## 0、前言

该文译自 WinCvs Daily Use Guide，原文撰写的时间可能比较早，相对于 WinCvs 的最新版，有些错误，主要是菜单的选择上。根据我个人的使用经验，增加了如何对文件或目录移动或改名（这主要在 UNIX 上完成，你也许需要一些 UNIX 的基本使用知识）。
CVS 受到了很多人的宠爱，希望也能受到你的宠爱。
——译者 Mirnshi

## 1、介绍

本文档介绍了WinCvs 1.0.x 客户端的日常使用。这不是介绍版本控制系统，也不是介绍 CVS，更不是介绍 WinCvs。这更适用于当你大概知道你要做什么，但又不十分清楚如何作。
假设你已经安装了 WinCvs，并且也已经配置好了与你要使用 CVS 库通讯。

## 2、术语

CVS 文档和 WinCvs 中的术语也许和其他的源码库系统不同。为了避免混淆，我们列举了最基本的术语简短列表。在翻阅文档的其它部分前，请首先熟悉这些术语。

*Checkout--检出*：一般用于描述从库中将一个整个模块的首次导出。
*Commit--提交*：将你的修改提交到库中。
*Export--输出*：指从库中将一个模块全部导出，但没有任何 CVS 管理文件。输出的模块没有 CVS 控制。
*Import--输入*：通常指通过提交一整个目录结构创建一个新的模块的过程
*Module--模块*：一个目录层。一个软件工程通常作为一单一模块存放在库中
*Release--发行版本*：整个产品的版本
*Revision--修订版*：单个文件的版本
*Tag--标签*：在一个开发的特定期对一个文件集给定的符号名
*Update--更新*：从库中取出其他人的修改。只更新本地的拷贝。

关于修订，发行版本，版本和标签的详细说明，参见 CVS 文档的第四章

## 3、创建新模块

将一个工程置于版本控制，在 CVS 术语中称作输入。从名字上就可以看出，在考虑版本控制前，你要为此作些准备工作。

对于一个工程的文件层次的重新构建，CVS 缺乏良好的支持。如果你在输入一个新的模块前花费一些时间来制定个计划，将会减少很多的麻烦。

输入操作的基本要求是有个“干净”的目录结构。“干净”的意思是不需要版本控制的文件都被移走了（如编译生成的文件，备份文件等等）。如果你的工程已经开始一段时间了，这就显得很重要。在目录中，你也许有些你不打算将它们置于版本控制下的文件，但是又想将他们放在那，这种情况下，你要在输入之前将它们移走，然后再移回来。

注意的是 CVS 认为空目录是不存在的。如果你想增加一个既不包含文件又不包含子目录的目录，你需要在其下创建一个哑文件。我们建议你创建一个名为 README.txt 的文件，其内容为对目录的简要说明。

一旦你想放到库中的文件都包含在了这个目录下以后，使用 WinCvs 用下面的步骤输入模块。

1) 选择 *Create->Import module* 菜单
2) 在弹出的文件对话框中，选择你要输入的工程目录的顶级目录，并确使文件夹图标打开。
3) WinCvs 然后试图辨别目录树中的任何二进制文件。如果弹出了筛选窗口，确保提及的文件被正确地设置了文本或二进制。
4) 在 *Import settings* 输入设置对话框，输入模块名。这将作为人们将来检出模块的顶级目录名。
5) 在 *Vendor tag* 厂商标记栏输入你的名字或公司的名字（没有空格）
6) 在 *Release tag* 发行版本标记栏输入 start
7) 点击 *OK* 按钮

WinCvs 然后将整个目录结构输入到 CVS 服务器中。注意在输入操作中对你的本地文件没有任何改变。也就是说，在输入后，你的本地拷贝没有置于版本控制下。在对你的版本控制源文件工作前，你需要作：

1) 移走你的原始目录，如用 Windows 的资源管理器将工程的顶级目录更名为 *.old。
2) 使用从 CVS 获取新模块一节中的步骤检出新模块的拷贝。
3) 还有拷贝回你移走的不重要的文件。

你现在就可以在版本控制上，开始项目工作了。

[ CVS 文档："使用 CVS 开始一个工程" ]

## 4、从 CVS 获取新模块

第一次从 CVS 服务器获得一个模块，被称作检出。从库中检出一个模块，会在本地创建构成模块的目录层。使用下面的步骤，执行检出：

1) 选择 *Create->Checkout module* 菜单。
2) 在弹出的文件对话框中，选择你要存放模块的目录。一个名为模块名的目录将会创建在你选择的目录下，所以通常你要选择存放所有工程的目录。
3) 一个检出设置对话框弹出后，在 *Enter the module name...*模块名输入框内，输入模块名。注意模块名区分大小写。
4) 选择 *Globals* 框
5) 根据项目组长告诉你的，设置检出的只读项 *Checkout read-only*
6) 点击 *OK* 按钮

如果你规定 *Checkout read-only* 检出应为只读，那么在编辑前，你要告诉 WinCvs 你要编辑的文件。这也许很麻烦，但是可以让其他开发人员跟踪知道谁正在编辑相应的文件。参见谁在编辑文件一节，获取更多的信息。

[ CVS 文档: "检出 –为了编辑检出源文件" ]

## 5、从 CVS 获取别人的修改

有时候，你也许要求将别人修改的合并到你本地使用的拷贝中。从服务器将修改取到本地的过程称为更新。可以更新单个文件、被选中的文件集，或者一整个目录层。执行下面步骤，完成更新。

1) 选择你要更新的目录或文件
2) 在选中的目录或文件上点击鼠标右键，选择 *Update selection* 更新菜单项
3) 确信 *Create missing directories that exist in the repository* 被选中
4) 点击 *OK* 按钮。

别人的修改将合并到你的文件中，并保留你对相同文件的任何修改。更新对库是没有影响的。

如果在更新时看到冲突的信息，请翻阅下一节

[ CVS 文档: "获取最新的文件" ]
[ CVS 文档: "update – 用同步方式或缺库中的工作目录树" ]

## 6、解决冲突

CVS 服务器偶尔在你从库中更新本地的文件时会报告冲突。当二个或更多的开发人员修改了同一个文件时，就会发生冲突。CVS 对你的工程一无所知，所以将解决冲突留给了开发人员。一旦发生了冲突，你就要打开有问题的文件，搜索以<<<<<<<.开始的行。冲突区有这样标记：

```
<<<<<<< 文件名
你的修改
=======
从库中合并的代码
>>>>>>> 修订版
```

你要判断代码应该是什么，作必要的修改，删除 CVS 的标记，然后将你的修改提交到库中。

[ CVS 文档: "冲突的例子" ]

## 7、向 CVS 提交你的修改

将本地的修改放到库中，被称作提交修改

1) 在提交前，你应该做一次更新，确保没有冲突。参见**从 CVS 获取别人的修改**
2) 选择你要提交的目录或文件
3) 在选择上点击鼠标右键，从菜单中选择 *Commit selection* 提交
4) 在 *Commit settings* 提交设置对话框中，输入日志信息。这步是可选的，但是强烈建议你花点时间简短描述你做的修改。
5) 点击 *OK* 按钮。

请注意，提交修改不会自动将你创建的新文件增加到库中。参见**增加文件和目录**，获得有关说明

[ CVS 文档: "提交 – 将文件提交到库中" ]

## 8、查看修改

WinCvs 可以查看文件和目录的状态、日志、区别等等。

1) 选择你要查看的目录或文件
2) 在选择上点击鼠标右键，选择 *Diff selection* 区别、*Log selection* 日志、*Status selection* 状态或 *Graph selection* 图示
3) 如果弹出对话框，输入必须的信息，然后点击 *OK* 按钮。

下面是几个命令的简要说明

*区别*

对本地拷贝与库中对应文件的任一修订版进行比较。也可以对库中的不同的修订版进行比较。输出是以"<"或">"开头的行，分别表明修订版和另外一个相比较的删除或增加。

*日志*

显示指定文件所有修订版（除非你限定了要显示的修订版）的日志信息、日期、标记、作者等等。

*状态*

显示指定文件的修改状态，如文件是本地修改还是库中修改。如果可能，还显示本地和库中的修订版号。

*图示*

一个很酷的特性，图形显示给定文件的修订版。当文件有一个或多个修订版分支时，特别有用。本地修订版标记为文档图标。借此，你可以选择二个修订版，作以比较（使用 shift 键选择）

[ CVS 文档: "区别 – 显示不同修订版间的区别" ]
[ CVS 文档: "日志 – 输出文件的日志信息" ]

## 9、对文件进行标记标签

在开发的一个指定阶段，对一个或多个文件进行标记标签，作为修订版的参照，被称为对这些文件标记标签。标签用于整个模块的特例是为了将来模块的当前状态可以重新构造。这种标签总是在工程发行或开始重大修改时进行

对一个或多个文件或目录标记标签，执行下面的步骤：

1) 选择你要标记标签的目录或文件
2) 选择 *Modify->Create a tag on selection*
3) 在 *New tag name* 新标签名输入框内输入标记（参见下面有关标记名的限制约定）
4) 点击 *OK* 按钮。

注意：CVS 对标签所包含的字符是非常严格的。一个标签必须以字母开头，后续多个字母，数字，减号和下划线。特别指出的是没有点号，也没有空格。如果你想在标签中包含版本号，请将点号换成减号。二个标签名作为 CVS 中的特有含义被保留了，“ HEAD ” 指在库中最新可用的的版本，而 “ BASE ” 指你最后检出到本地的修订版。

[ CVS 文档: '标签 – 符号化的修订版' ]

## 10、增加文件和目录

当你创建了想要放到库中的文件时，你要告诉 CVS 要做相应处理。如果，目录包含不在 CVS 控制下的文件，你需要增加文件前增加目录。增加文件或目录，要：

1) 选择你要增加的目录或文件
2) 选择 *Modify->Add selection* 或 *Add selection binary* 菜单，对于非文本文件，选择二进制，免得文件会被 CVS 破坏掉
3) 文件只是被标记为添加，你要通过提交将其增加到库中。如果你不知道如何提交，参见向 CVS 提交你的修改。

如果你要增加一个目录层，而不是几个文件，上面的技术有点麻烦，因为增加操作不支持子目录。这种情况下，你要使用输入，就像创建新模块。

1) 选择 *Create->Import module* 菜单
2) 在弹出的文件对话框中，选择你要增加的目录，并使这个文件夹打开。
3) WinCvs 然后试图辨别目录树中的任何二进制文件。如果弹出了筛选窗口，确保提及的文件被正确地设置了文本或二进制。
4) 在 *Import settings* 输入设置对话框，输入模块名和目录名。如果模块名叫 MyProject,你要输入的子目录为 source/utils,在 *Select the module name...*模块名输入框中输入：MyProject/source/utils
5) 在 *Vendor tag* 厂商标记栏输入你的名字或公司的名字（没有空格）
6) 在 *Release tag* 发行版本标记栏输入 start
7) 点击 *OK* 按钮。

WinCvs 然后将整个目录结构输入到 CVS 服务其中。注意在输入操作中对你的本地文件没有任何改变。也就是说，在输入后，你的本地拷贝没有置于版本控制下。在对你的版本控制源文件工作前，你需要作：

1) 移走你的原始目录，如用 Windows 的资源管理器将工程的顶级目录更名为 *.old。
2) 在 WinCvs 中，选择你增加的目录，点击鼠标右键，选择 *Update selection* 更新
3) 确信 *Create missing directories that exist in the repository* 被选中。
4) 点击 *OK* 按钮。

WinCvs 就提供了最新置于版本控制下的目录层

[ CVS 文档: "增加,删除,改名文件和目录" ]

## 11、删除文件和目录

要删除文件，你首先要确定删除的时间，然后提交更改。

1）选择你要删除的文件
2）选择 *Modify->Remove selection* 菜单
3）文件只是被标记为删除，你还要通过提交将其从库中删除。如果你不知道如何提交，参见向 CVS 提交你的修改。

文件就会从库中被删除了。注意文件没有物理上删除，只是被标记为"消亡"。在这种方式下，如果你选择检出模块的老的修订版，仍可能恢复出删除的文件。

删除目录时另外一个话题。当你更新一个目录的父目录时，CVS 会删除空的目录。如果你想删除一个空目录，操作如下：

1）选择你要删除的空目录的父目录
2）点击鼠标右键，选择 *Update selection* 更新菜单
3）选择 *Globals* 框
4）确定 *Prune (remove) empty directories* 被选中（这是缺省项）
5）点击 *OK* 按钮。

如果一个目录中文件先前已被从本地和库中删除，这个空目录就会被删除。

[ CVS 文档: "增加,删除,改名文件和目录" ]

## 12、对文件和目录移动或改名

对文件和目录移动或改名的操作在 CVS 中是不允许的。这是 CVS 的一个不足之处。你需要合并删除和增加操作来模拟移动或改名。参见增加文件和目录以及删除文件和目录。

译者注：
在实际操作中,用户可以直接到 CVS 服务器上，在库目录下，直接对相应的文件作移动和改名操作。新文件名要符合 CVS 库文件的命名规则。一般情况下，不建议用户直接到 CVS 服务器上直接修改。修改时，注意文件的属主和存取权限。

[ CVS 文档: "移动或改名文件" ]

## 13、分支

版本控制系统的一个特性是，能够将修改隔离到一个分离的开发线上。这条线称为分支（参见 CVS 文档中的What branches are good for什么分支是有益的），通过下面的步骤，创建一个分支

1） 选择你要分支的目录或文件
2） 选择 *Modify->Create a branch*
3） 在 *New branch name* 新分支名输入框内，输入你要求的分支的标记名。参见对文件进行标记标签，了解标签中的合法字符
4） 选中 *Check that the files are unmodified before branching*
5） 点击 *OK* 按钮。

现在在库中就按照指定的名字创建了一个新的分支。分支只是在库中创建了。要在新创建的分支上工作，你还要按照选择一个分支工作一节中的说明去做一些工作。

[ CVS 文档: "分支和合并" ]

## 14、选择一个分支工作

要在一个分支上而不是在缺省的开发线上工作，你要将你本地的拷贝接到分支上。这需要确信一些操作如更新、提交等等是在分支上而不是在开发的主线上。

将本地拷贝转移到另外分支上，操作如下：

1) 选择工程的顶级目录（如果你准确的知道分支内的目录和文件，也可以分别选择）
2) 在选择上点击鼠标右键，选择 *Update selection* 更新菜单
3) 确信 *Create missing directories that exist in the repository* 被选中。
4) 选择 *Sticky options* 粘着项框
5) 选中 *Retrieve rev./tag/branch*
6) 在 *Retrieve rev./tag/branch* 输入框，输入你要切换到的分支标签名
7) 点击 OK 按钮。

WinCvs 就会作需要的更新，更新你工作着的拷贝并转移到预期分支上。更新操作将增加或删除文件。
CVS 对分支的文件标记了粘着标签。你可以通过对文件执行状态命令，查看这些粘着标签。参见查看修改。要删除粘着标签，返回到开发主线上，执行返回开发主线一节描述的步骤。

[ CVS 文档: "访问分支" ]

## 15、从分支中合并

当你对你在分支上作的修改很满意时，你也许想开发主线也能获得这些修改。将一个分支上的修改合并到另一个分支上，被称为合并。要合并一个分支，要执行：

1) 将本地拷贝移到你要将修改合并到的分支上。参见选择一个分支工作或返回开发主线
2) 选择工程的顶级目录（如果你准确的知道分支内的目录和文件，也可以分别选择）
3) 在选择上点击鼠标右键，选择 *Update selection* 更新菜单
4) 确信 *Create missing directories that exist in the repository* 被选中。
5) 选择 *Merge options* 合并框
6) 选中 *Only this rev./tag*
7) 在 *Only this rev./tag* 输入框内，输入你要合并的分支标签名
8) 点击 *OK* 按钮。

这个分支上的任何修改就会合并到你的本地拷贝中。你可能还想将合并后的文件提交回库中，参见向 CVS 提交你的修改

重要提示：上面提到的合并将合并从分支起始的修改。如果你是第二次操作（在最后合并又作合并），从分支开始的合并并不是你所期望的，多半会让你陷入困境。为了避免这个问题，你应当在每次合并后，为分支指定新的标签，为后来的合并命名分支时要使用新的标签。

[ CVS 文档: "合并整个分支" ]

## 16、返回开发主线

**如果你想停止分支上的工作，将本地拷贝转回到开发主线上，你要通过 WinCvs 删除所有的粘着标签。要删除粘着标签更新本地拷贝，回到开发主线等，执行下面的操作：**

1) **选择工程的顶级目录（如果你准确的知道分支内的目录和文件，也可以分别选择）**
2) **在选择上点击鼠标右键，选择** *Update selection* **更新菜单**
3) **确信** *Create missing directories that exist in the repository* **被选中。**
4) **选中** *Reset any sticky date/tag/'-k' options*
5) **点击** *OK* **按钮。**

**WinCvs 就会更新你的本地拷贝，以使符合当前的开发主线。曾经工作的分支仍然存在库中。无论何时你都可以返回到这个分支上，参见选择一个分支工作**

[ CVS 文档: '粘着标签'' ]

## 17、谁在编辑文件

查询 CVS 谁在编辑一个文件，是可以的，但是只有开发人员通知 CVS 他们打算要编辑一个文件时才可以。这种方式被称为 IR CVS 方式，而且如果要使用这种方式，就要在工程开始时就决定好。

要使用 IR CVS 方式，开发人员要以只读方式检出模块。可以在每个 WinCvs 对话框的 Gloabals 框中选中 Checkout read-only 做到这点。

要编辑一个文件，执行下面的步骤

1） 选中你要编辑的文件
2） 选择 *Trace-> Edit selection* 菜单，只读标记就会被删除，并且 CVS 服务器会登记你是指定文件的编辑者
3） 编辑文件
4） 按照向 CVS 提交你的修改的说明提交文件。

当你提交文件后，CVS 服务器就假定你不再编辑它了，WinCvs 又将文件标记为只读。如果你不作任何修改，提交不会通知 CVS 服务器你已完成了编辑。这种情况下，你要执行下面的操作

1） 选中你标记为编辑的文件
2） 选择 *Trace-> Unedit selection* 菜单，

现在，回到原来的问题 :“ 谁在编辑文件 ”。要列出一个文件的编辑者，执行：

1） 选中你关注的文件
2） 选择 *Trace-> Editors of selection* 菜单，

WinCvs 就会列出文件的编辑者

[ CVS 文档: "多位开发人员" ]

## 18、附录 A：常见问题及解答

1) **在更新时，难以理解的字母的含义是什么？**

   **参见 CVS 手册中的**update output**更新输出**

2) **服务器的时钟错了，标签和日志显示错误的时间**

   **这是个特性，CVS 总是使用 UTC 时间**

## 19、附录 B：CVS 与微软的 Visual Source Safe 有何不同

CVS 和 WinCvs 与 Visual Source Safe (VSS)在很多处都不同。最明显的区别是 CVS 不需要用户锁定他们正在工作的文件，而 VSS 对此是缺省。实际上，CVS 文档甚至鼓励用户不要使用文件锁。几个人在同一时间修改同一文件几乎很少发生。CVS 通常能合并他们的修改。如果二个或更多的开发人员改变了相同的几行，CVS 会报告冲突，在文件中插入标识，留给开发人员决定如何作。这样的冲突是非常罕见的，而且通常发生在开发人员间缺乏交流时才会出现。(如：二个人试图修订相同的问题)

另一个重要的不同是 VSS 给你的一个服务器的视点，而 WinCvs 给出的是客户端的视点。实际中的含义就是，不象 VSS，WinCvs 不告诉你库中的修改，直到你要做更新。或明确地查询被选择的文件状态。要在 WinCvs 中反映出自你最后检出以来的修改报告，需要更新或提交

CVS 不同于 VSS 的可视化。当使用 WinCvs 时，就很清楚了。在一些人决定使用 GUI 前，CVS 一直使用命令行操作已经很多年了。我们中的一些人喜欢 WinCvs 的简单的方法，这给有经验的用户一种控制的感觉。其他人也许要讨厌这点，如果你是其中一员，那就自由的改善这个程序，并将你的修改回赠给社会。